2702Task2String

May 12, 2025

1 Single line comment

[8]:
```
letter = 'P' # A string could be a single character or a bunch of texts
print(letter)
```

```
P
```

[10]:
```
print(len(letter))
```

```
1
```

[13]:
```
Greeting="Hello World"
print(Greeting)
```

```
Hello World
```

[14]:
```
print(len(Greeting))
```

```
11
```

[18]:
```
Sentence="I Hope you are enjoying 30days of Python Challenge"
print(Sentence)
```

```
I Hope you are enjoying 30days of Python Challenge
```

[20]:
```
print(len(Sentence))
```

```
50
```

1.1 Multiple String

[22]:
```
multiline_string = '''I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.'''
print(multiline_string)
```

```
I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.
```

[23]:
```
# Another way of doing the same thing
```

[24]:
```
multiline_string = """I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python."""
print(multiline_string)
```

```
I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.
```

2 String Concatenation

[32]:
```
first_name = 'Asabeneh'
last_name = 'Yetayeh'
space = ' '
```

```
Asabeneh Yetayeh
```

[35]:
```
full_name=first_name+space+last_name
print(full_name)
```

```
Asabeneh Yetayeh
```

[36]:
```
# Checking length of a string using len() builtin function
print(len(first_name))  # 8
print(len(last_name))   # 7
print(len(first_name) > len(last_name)) # True
print(len(full_name)) # 16
```

```
8
7
True
16
```

2.1 Unpacking characters

[42]:
```
language="python"
a,b,c,d,e,f=language
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

```
p
y
t
h
o
n
```

2.2 Accessing characters in strings by index

[52]:
```
language='Python'
first_letter=language[0]
print(first_letter)
Second_letter=language[1]
print(Second_letter)
last_index=len(language)-1
last_letter=language[last_index]
print(last_letter)
```

```
P
y
n
```

[ ]:
```
# If we want to start from right end we can use negative indexing. -1 is the␣
↪last index
```

[53]:
```
language = 'Python'
last_letter = language[-1]
print(last_letter) # n
second_last = language[-2]
print(second_last) # o
```

```
n
o
```

2.3 Slicing

[56]:
```
language='python' # it will give result of 0,1,2 positions 3 is not included
first_three=language[0:3]
last_three=language[3:7]
print(first_three)
print(last_three)
```

```
pyt
hon
```

2.3.1 Another way to get result

[59]:
```
last_three=language[-3:]
print(last_three)
```

hon

[61]:
```
last_four=language[-4:]# : will give the series of letters till that number
(positive then till that number-1 and if the number is - like -4: then gives␣
 ↪last four numbers
 print(last_four)
```

thon

2.3.2 Skipping character while splitting Python strings

thisw

[ ]:

[ ]:

[ ]: